Лашко Никита ПМИ-3

## Кроссплатформенная мобильная разработка с помощью React-Native

React Native – это open-source платформа для создания мобильных приложений, созданная Facebook в 2015 году. В ее основе лежит фреймворк для веб-разработки React.js. React Native используется для разработки приложений под Android, iOS, UWP, позволяя разработчикам использовать React вместе с нативными возможностями платформы.

Он позволяет создавать мобильные приложения, используя при этом только JavaScript с такой же структурой, что и у React, только рендериться они будут не в HTML, а в нативные контролы операционной системы, под которую будет собрано наше приложение. Это дает возможность составлять многофункциональный мобильный UI с применением декларативных компонентов.

## Базовая идея, возможности и особенности данной технологии

Поскольку React Native по большей части основан на фреймворке для разработки веб-интерфейсов React.js, то в первую очередь рассказать про него.

React.js имеет следующие особенности:

1. React приложение состоит из компонентов – инкапсулированных блоков функциональности.
2. Свойства передаются от родительских компонентов к дочерним. Компоненты получают свойства как множество неизменяемых значений, поэтому компонент не может напрямую изменять свойства, но может вызывать изменения через callback функции.
3. Структура веб-страницы представлена с помощью виртуального DOM, который представляет собой легковесную копию обычного DOM.
4. React использует JavaScript XML (JSX) – расширение синтаксиса JavaScript, которое позволяет использовать похожий на HTML синтаксис для описания структуры интерфейса.
5. У каждого компонента существуют методы жизненного цикла, которые позволяют разработчику запускать код на разных стадиях жизненного цикла компонента.

React Native стремится внести концепции построения пользовательских интерфейсов с помощью React в разработку мобильных приложений и задействовать наилучшие возможности мобильной разработки и разработки браузеров.

Он поощряет совместное использование кода на разных платформах (есть компоненты, предназначенные как для iOS, так и для Android-устройств), позволяет писать нативный код там, где это целесообразно, и компилируется в нативное приложение – все это применяя множество схожих идиом, известных в React.

React Native применяет базовую библиотеку React для создания компонентов, но другой набор библиотек – для поддержки рендеринга приложения на нативной платформе и взаимодействия с базовой платформой (события касания, геолокация, доступ к камерам и т. д.).

React.js использует виртуальную DOM для создания лучшего UI, React Native же использует собственные API-интерфейсы для отображения частей пользовательского интерфейса, которые могут быть повторно использованы как на платформах iOS, так и на Android. А также Java API для визуализации компонентов Android и API Objective-C для написания компонентов iOS.

Кроме этого, используется JavaScript для остального кода и индивидуализации приложения для каждой платформы. Это дает возможность React Native возможность повторного использования компонентов и кодов.

Многие API, специфичные для React (например, используемые в жизненных циклах компонентов, состоянии, свойствах и т. д.), применимы и для React Native. Но каждая нативная платформа реализует собственные различные API для работы с сетью, разметкой, геолокацией, управлением ресурсами, сохранением, событиями и другими важными областями.

React Native стремится импортировать некоторые знакомые API из мира, ориентированного на браузеры, например API Fetch для сетевого взаимодействия и API Flexbox для компоновки.

Также React Native предоставляет события, но они более специфичны для мобильных платформ (например, onPress).

## Особенности мобильной разработки

Из особенностей непосредственно самой разработки, можно выделить следующее:

1. Как было сказано ранее, мы не ограничены только языком JavaScript, и можем писать нативный код (например, на Java) там, где это необходимо.
2. Не нужно тратить много времени на сборку, чтобы протестировать новые функции приложения, поскольку React Native обладает функцией «горячей перезагрузки». Благодаря этой функции приложение перезагружается каждый раз, когда происходит сохранение исходных файлов приложения.
3. В React Native вызов API, рендеринг изображений по URL и другие процессы очень просты. Когда данные поступают из API на платформе Android, они сначала преобразуются в POJO-модель и лишь затем используются в элементах UI. А вот данные JSON, полученные в React Native, удобны для JavaScript и могут напрямую использоваться для предпросмотра UI. Это позволяет облегчить веб-интерфейс для GET или POST-запросов от REST API.
4. Отсутствие визуального редактора интерфейса – всё делается в коде с помощью JSX-разметки.
5. Отсутствие UIViewController и Activity, они есть под капотом, но напрямую взаимодействовать с ними не получится. UI-элементы сами меняют состояние / вид при изменении входных параметров.

React Native обладает своими плюсами и минусами, отлично подходит для разработки небольших приложений или создания минимально жизнеспособного продукта. Однако встать на одном уровне с нативной разработкой он на данный момент не в состоянии.

Несмотря на то, что технология и сообщество React Native по-прежнему находится в стадии становления, это серьезный инструмент для разработки кросс-платформенных мобильных приложений, с которым приходится считаться.

## Источники

1. Томас Марк Тиленс React в действии. – СПб.: Питер, 2019.
2. <https://habr.com/ru/company/plarium/blog/458118/>
3. https://habr.com/ru/company/qlean/blog/416097/
4. <https://www.purrweb.com/blog/ru/react-native-plyusy-i-minusy/>
5. <https://reactnative.dev>
6. <https://medium.com/devschacht/create-devschacht-app-part-1-db9570cf3f9b>